Geheimnisse in OpenStack mit Barbican verwalten

Streng geheim

Martin Gerhard Loschwitz

Barbican hilft, in OpenStack-Clouds Passwörter, Crypto-Keys und SSL-Zertifikate zu verwalten. Wie Cloud-Anwender Barbican effizient nutzen, verrät dieser Artikel.

Wer virtuelle Umgebungen in öffentlichen oder privaten Clouds betreibt, muss Geheimnisse wie Passwörter, SSH-Schlüssel oder SSL-Clientzertifikate sicher verwahren können. Barbican ist ein OpenStack-Dienst, der eine REST-Schnittstelle bietet und genau dafür gemacht ist.

Ist Barbican noch nicht installiert, muss man dies nachholen, indem man zunächst in Keystone die entsprechenden API-Endpunkte für Barbican hinterlegt und danach die Barbican-Komponenten einspielt. Zuvor legt man in MySQL eine Datenbank für Barbican an, in der es später seine Metadaten deponiert. Schließlich installiert der Administrator die zu Barbican gehörenden Pakete, auf Ubuntu etwa *barbican-api, barbican-keystone-listener* und *barbican worker*. Eine ausführliche Anleitung hierzu, die auch verschiedene OpenStack-Distributionen abdeckt, stellt das Barbican-Projekt im Netz zur Verfügung (siehe ix.de/ix1804132). Es existieren auch Module für Puppet sowie ein Playbook für Ansible, um Barbican automatisiert einzuspielen.

Barbican für Cinder und Nova

Grundsätzlich beherrscht sowohl die OpenStack-Virtualisierungskomponente Nova als auch die Volume-Komponente Cinder das Anlegen und Verwalten verschlüsselter Volumes. Damit das funktioniert, müssen beide Dienste so konfiguriert sein, dass sie auf Barbican zurückgreifen. In *nova.conf* sind dazu die Konfigurationsabschnitte „keymgr“ und „barbican“ entsprechend zu bearbeiten; in *cinder.conf* fügt der Admin eine „barbican“-Sektion hinzu, in der er auf Barbican verweist. Die Cinder-Autoren erklären in ihrer Dokumentation ausführlich weitere Setup-Details.

Nun lassen sich Geheimnisse in Barbican ablegen. Etwa so:

```
openstack secret store -name ixpasswort - 7
                                paylog ganzgeheim
```

Damit das funktioniert, muss auf dem System das Modul *python-barbicanclient* vorhanden sein. „name“ definiert den Namen, unter dem Barbican den Eintrag in der Datenbank ablegt. Die Payload ist das eigentliche Passwort. Barbican antwortet mit einer längeren JSON-Antwort, die der Client lesbar darstellt:

```
Field     Value
Secret href
http://10.42.0.1:9311/v1/secrets/5395dc32- 7
                    1d54-11e8-b071-00215acd73e2
Name      ixpasswort
Created   None
Status    None
Content types          None
Algorithm  aes
Bit length 256
Secret type            opaque
Mode      cbc
Expiration None
```

Ein wichtiger Teil der Antwort ist das Feld „Secret href“ – quasi eine ID, mit der das Passwort sich wieder aus Barbican auslesen lässt:

```
openstack secret get "HREF-Link"
```

Möchte man die Payload erneut auslesen, hängt man an das Kommando den Parameter *--payload* an.

Neben der Möglichkeit, geheime Inhalte wie Passwörter in Barbican abzulegen, ist der *order*-Befehl sicher ein Highlight der Software. Dabei handelt es sich um eine Art Lieferdienst für Passwörter: Der Befehl

```
openstack secret order create secret
```

generiert ein Passwort und speichert es in Barbican. Wieder gibt der Client HREF-Definitionen aus, allerdings deren zwei: Die erste gehört zur Bestellung, die zweite zum Passwort, das tatsächlich in Barbican liegt. Das auf diese Weise angelegte Passwort unterscheidet sich übrigens nicht von einem, das der Nutzer händisch in Barbican speichert.

Zertifikate generieren

So kann das Tool auch SSL-Zertifikate generieren. Dazu muss man in der Konfiguration ein CA-Zertifikat zum Signieren der SSL-CSR-Dateien hinterlegen. Dann geht der Rest leicht:

```
openstack secret order create --subject-dn 7
                    "WERT" --ca-id "ID" certificate
```

Wieder sind HREF-Verweise Teil der Antwort; Barbican enthält nun das Zertifikat nebst Schlüssel. Ansehen kann man sich das mit dem Befehl *openstack secret ca list*.

Barbican versteht sich als eine Art Passwortmanager: Wer in einer virtuellen Umgebung mit Passwörtern hantiert, legt diese in Barbican zentral und per API abfragbar ab. Es empfiehlt sich, auch aus virtuellen Maschinen heraus die Barbican-Funktionen zu nutzen: Braucht man in einer VM ein Passwort, kann man den OpenStack-Barbican-Client nutzen, um es direkt aus einer VM heraus anzulegen oder abzufragen. Ähnliches gilt, wenn man etwa für einen Webserver in einer VM ein passendes SSL-Zertifikat benötigt.

Quintessenz: Barbican bietet eine Reihe von Funktionen, und aus Security-Sicht empfiehlt es sich, sie zu nutzen. Denn das ist auf jeden Fall besser, als Passwörter statisch in VM-Images abzulegen. (js@ix.de)

Martin Gerhard Loschwitz

ist Public Cloud Architect bei T-Systems und beschäftigt sich vorrangig mit OpenStack, Ceph und Kubernetes.

Alle Links: ix.de/ix1804132